Qualitative Assessment of Access Control in a Database Management System

Isabel Sofia Brito
ESTIG/IPBeja
Beja, Portugal
isabel.sofia@ipbeja.pt

Luís Sobral
ESTIG/IPBeja
Beja, Portugal
luis.sobral@ipbeja.pt

***Abstract*—This paper presents a qualitative assessment of access control in database management system to guide those who wish to implement a discretionary or/and non-discretionary access control model and need some support to choose the access control in database management system (DBMS) best suited to their security requirements. To accomplish this we apply the core concepts related to access control models, and the metrics in NISTIR 7874. The result of this work shows how the database management system chosen, MS SQL Server 2012 supports the core concepts and the most popular access control models: RBAC, DAC and MAC, all these based on NIST 7874 metrics.**

Keywords— access control models, DBMS, NISTIR 7874

I. INTRODUCTION

This paper shows the result of a qualitative assessment of access control in MicroSoft (MS) SQL Server 2012, a commercial database management system (DBMS). This assessment is based on access control concepts, such as separation of duty and least privilege [3] and, the most popular access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-Based Access Control (RBAC). To accomplish this, a set of experiments was elaborated in MS SQL Server 2012 and a "real-world" database was used. The document "Guidelines for Access Control System Evaluation Metrics" [8] published by the National Institute of Standards and Technology (NIST) has also been applied to the experiments and thus considers other aspects in the assessment, such as ease of implementation of control.

The remainder of the paper is organized as follows. Section II, background, is composed of three parts: the first part presents the NIST report and the quality metrics used in this paper. The second part discusses the main characteristics of the discretionary, mandatory, and role-based access control models, and last part describes some basic concepts about access control in MS SQL Server 2012. Section III describes and analyzes the DAC and RBAC access control supported by MS SQL Server 2012. Section IV analyzes the access control supported by MS SQL Server 2012 according to NIST defined evaluation metrics. The metrics are also used to compare DAC and RBAC access control. Section V describes the related work. Finally, in Section VI some concluding remarks are presented.

II. BACKGROUND

A. NIST Assessment Criteria

In 1992 NIST initiated its work on study of the commercial and government organizations, and elaborated several reports to help enterprises within industry and government to select the best access control for its systems [2][4][8][10]. The report in [8] presents properties for quality metrics of access control systems using concepts described at the beginning of this section. The metrics, built on those originally defined in [2], can be used when considering and comparing the properties for current configuration or future expansion of an access control system. The properties to be evaluated are divided into four categories according to organization's operational needs: i) administration is the main consideration of cost, ii) enforcement capabilities are the requirements for access control applications, iii) performance is a major factor for access control usability, and iv) support functions allow an access control system to utilize and connect to related technologies. For each property, a list of functions needs to be considered. The information provided for each function in these four categories follows the same format. For each function, a list of metric items to evaluate is presented with supporting descriptions, among other information not used in this work.

This paper evaluates the access control mechanisms of MS SQL Server 2012 on the basis of these metrics in practical experiments. Considering our case study not all metrics can be applied, so we have selected the most appropriate NIST access control metrics for the evaluation of access control in DBMS, as described in Section IV.

B. Access Control Models

Access control models bridge the gap between policy and mechanism, i. e. models can be promoted for their support of policy, and mechanisms can be designed for their adherence to the properties of the model [2]. This work is based on the three well known access control models: DAC, MAC and RBAC.

As shown in [3] and based on Trusted Computer System Evaluation Criteria (TCSEC), DAC is an access control that permits users to allow or disallow other users access to objects under their control. In database systems context, DAC mechanism allows users to grant or revoke access to any of the objects under their control without the intercession of a system administrator. DAC mechanisms are included in the SQL

978-1-4673-8611-1/15/$31.00 ©2015 IEEE

standard, i. e. the creator of an object in a SQL database is its owner with the ability to grant other users access to that object. MAC is defined in TCSEC as follows [3]: "A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to information of such sensitivity." In DBMS point of view, MAC is not supported directly in SQL. RBAC is defined as: "Access to computer system objects is based on a user's role in an organization. A role was seen as a job or position within an organization."[3]. A role is essentially a collection of permissions, and all users receive permissions only through the roles to which they are assigned. Controlling all access through roles therefore simplifies the management and review of access controls. In DBMS context the RBAC features are: i) user role assignment; ii) support for role relationships and constraints, and iii) assignable permissions [6].

C. Access Control on MS SQL Server 2012

MS SQL Server 2012 DBMS was chosen for the assessment to present to the reader a set of access control models capabilities in a commercial product. MS SQL Server 2012 is one of the most influential and widely spread DBMS. SQL Server 2012 uses terms like users, roles, and permissions and, new words like principals, schemas, and securables [5].

The database user principal is related to DAC model and the database role principal is related to RBAC model.

As mentioned in Section II.B, SQL does not support MAC concepts, so SQL Server 2012 does not have natively tools that support this access control model. There is nevertheless a project called SQL Server Label Security Toolkit.

III. Access Control Experiments and Analysis

This section describes and analyzes the DAC and RBAC access control supported by MS SQL Server 2012. The experiments are constructed using AdventureWorks2012 database. This database is based on a fictitious multinational manufacturing company. The company manufactures and sells metal and composite bicycles to several commercial markets. For the study we will create five logins on the server and five users on AdventureWorks2012.

A. Practical Experiments to Verify DAC Mechanisms

As described before, the practical experiments are performed in MS SQL Server 2012 using database users created in AdventureWorks2012 database. These users have the possibility to grant, revoke or deny permissions on their objects to other users, which are mandatory features of a discretionary access control model. To identify these features eight experiments were implemented. For the practical experiment of access control model we will also use schemas. For example, User1 is owner of HumanResources schema, and this schema contains several objects of AdventureWorks2012 database.

Experiment 1: This experiment verifies the user's permissions (User1 in this case), when he/she is created in the database. The result shows that when a user is created he/she has no permission to execute any action in the database, supporting the least privilege principle. So, database user has no permission when he/she was created, having the permissions to be assigned by the administrator of the DBMS or by the owner of the object on which permissions will be given.

Experiment 2: The goal of this experiment is identify how many steps are needed to grant, revoke, and deny permissions to a user. As previously stated, MS SQL Server 2012 has a way of grouping objects into schemas, which makes easier to assign permissions on multiple objects to a given user. The experiment shows that the number of steps depends on the existence of a schema; i. e. a schema helps the administrator to assign permissions on multiple objects.

Experiment 3: The experiment's goal is to verify if a user can assign permissions to other users, even when he/she is not the owner. In MS SQL Server 2012 this is possible using WITH GRANT clause.

Experiment 4: The goal of this experiment is to revoke the permissions previously assigned to other users. According with DAC model, only the owner of the object can revoke permission. For WITH GRANT permission two steps are need and, one to revoke or deny grantable privileges, using CASCADE option.

Experiment 5: The experiment's goal is to verify if DENY permission takes precedence over GRANT permission. The result shows that DENY permission has precedence over GRANT permission. Notice, this precedence is also useful to solve conflicts, since two different users (for example, User1 and User2) give GRANT and DENY permissions on a given table to another user (User3) , the conflict is "automatically" solved because DENY has precedence over GRANT.

Experiment 6: The goal's experiment is verifying the delegation of control administration between users. MS SQL Server 2012 permits the administration delegation between users using GRANT CONTROL option.

Experiment 7: The goal of this experiment is verifying how MS SQL Server 2012 supports access control auditing and other type of access control information. To accomplish this, MS SQL Server 2012 has two audit levels: SQL Server Audit and Database Audit Specification. Using the last one is possible to audit the activity of a user. It is also possible visualize the information, using fn_ger_audit_file() a MS SQL Server function. MS SQL Server 2012 has tools that allow a user to visualize its permissions, such as fn_my_permission table. The administrator can visualize the permission on an object using the stored procedure sp_helprotect or on a given user using sys.database_permissions table.

Experiment 8: The goal's experiment is to verify the relationship between stored procedures and discretionary access control. A user can execute the stored procedure with success, for example SELECT, because he/she has permissions to do so despite the deny permission to perform SELECT. So, a user should be very careful when assigning execute permissions of its stored procedures to another user.

B. Practical Experiments to Verify RBAC Mechanisms

As described before, the practical experiments are performed in MS SQL Server 2012 using database users and roles created on AdventureWorks2012 database as well as the db_owner and db_securityadmin fixed database roles. Notice, members of the db_owner and db_securityadmin database roles can manage fixed database role membership, and every database user belongs to the public database role (for more information see [5]). When a user has not been granted or denied permissions on a securable object, the user inherits the permissions granted to public on that object. These elements are mandatory to implement role-base access control model using the experiments described below.

Experiment 1: This experiment verifies the user's permissions, when he/she is created in the database. The result shows that the public database role has no permission to execute any action in the database, supporting the least privilege principle; any permission has to be assigned by the administrator of the DBMS, for example.

Experiment 2: The goal of this experiment is identify how many steps are needed to grant, revoke, and deny permissions to a user. MS SQL Server has fixed and a non-fixed roles, so if the fixed database roles are used, only one step is necessary: authorize the user to be member of the fixed role. Furthermore, if a non-fixed role already exists, only one step is necessary: authorize the user to be member of the non-fixed role. If there is no database role to support a specific access control goal, a new database role can be created. Afterwards, we can authorize the users to be members of the newly created role. So, we can conclude that the number of steps depends on the existence of the database role. Because a role is a collection of permissions, if you need to change the permission, this should be done at role level, and all users "receive" this changes through the roles to which they are assigned.

Experiment 3: The experiment's goal is to verify if a user can assign permissions to other users, even when he/she is not the owner. In MS SQL Server 2012 this is possible using WITH GRANT. Notice that the permissions can be granted in cascading, so the owner of the object must be very careful when assigning permissions with the WITH GRANT option.

Experiment 4: The goal of this experiment is to revoke the permissions previously assigned to other users. The result of this experiment shows that only a role member can REVOKE permission given by the role.

Experiment 5: The experiment's goal is to verify if DENY permission takes precedence over GRANT permission. Concluding, DENY permission has precedence over GRANT permission. This precedence is also useful to solve conflicts, since the conflict is "automatically" solved because DENY has precedence over GRANT.

Experiment 6: The goal's experiment is to verify the delegation of control administration between users. MS SQL Server 2012 permits the delegation of control administration whenever a user was authorized to be member of a role with administrative permission. Notice, adding users to administrative role could enable unintended privilege escalation.

Experiment 7: The goal of this experiment is verifying how MS SQL Server 2012 supports access control auditing and other type of access control information. As described before, MS SQL Server 2012 has two audit levels: SQL Server Audit and Database Audit Specification. Using the last one is possible to audit the activity of a user. This could be important in access control because we can audit any action related with, for example, database role members. It is also possible to visualize this information, using fn_ger_audit_file() a MS SQL Server function. MS SQL Server 2012 has tools that allow a user to visualize its permissions. One of these tools was explained in Section III. A., the fn_my_permission table.

Experiment 8: The goal's experiment is to verify the relationship between stored procedures and role based access control. The result is that a user can execute SELECT on a table using the stored procedure, because he/she has permissions to do so despite the DENY SELECT permission on a table. So, a user should be very careful when assign execute permissions on stored procedures to another user.

IV. Access Control Evaluation Based on NIST Metrics

This section analyzes the access control supported by MS SQL Server 2012 according to NIST defined evaluation metrics. The metrics are also used to compare DAC and RBAC access control supported by MS SQL Server 2012. We have selected the following NIST 7874 properties for the evaluation of access control according to the practical experiment categories described before: i) administration and ii) enforcement capabilities. For each property, a list of functions needs to be considered. The analysis is presented for each item of the selected functions. The scale applied to the items is the following.

- Full support - When all function items are supported by the DBMS;
- Partial support - When some function items are supported by the DBMS;
- Not support - When none function items are supported by the DBMS;
- Not applicable – When function items cannot be measured, because they are not at the database level.

The items for the auditing function from administration category are analyzed in Table I.

TABLE I
AUDITING ITEMS

Metric Items to Evaluate	Analysis
Does the AC system log system failure?	Not applicable in this case study. MS SQL Server supports this item at server level.
Does the AC system log denied access requests?	Full support according with experiment 7 from Sections III.A and III.B
Does the AC system log granted access requests?	Full support according with experiment 7 from Sections III.A and III.B
Does the AC system provide additional log functions required by the organization?	Not applicable in this case study. MS SQL Server supports this item at server level.
Does the AC system provide additional log functions required by the organization?	Not applicable in this case study. MS SQL Server supports this item at server level.

According with the analysis, the auditing function is fully supported by MS SQL Server 2012.

The items in the privileges/capabilities discovery function, from administration category, are analyzed in Table II.

TABLE II
PRIVILEGES/CAPABILITIES DISCOVERY ITEMS

Metric Items to Evaluate	Analysis
Does the system provide query/display for (constrained) privileges/capabilities discovery?	Partial support according with experiments 7 and 8 from Sections III.A and III.B. A user with permission to execute a stored procedure can make the action contained therein, without being shown that this user is allowed, or not, to perform that action.
Does the system provide graphic display?	No support.

According to the analysis previously achieved, the privileges/capabilities discovery function is partial supported by MS SQL Server 2012. Notice, some items for this function is not applicable, as we said before, for example, "Does the system provide query/display for AC system states discovery for (constrained) privileges/capabilities?"

For the ease of privilege assignments function from administration category, the items are analyzed in Table III.

TABLE III
EASY OF PRIVILEGE ASSIGNMENTS ITEMS

Metric Items to Evaluate	Analysis
How many steps are required for assigning/changing/removing privilege or a capability to a subject/subject group?	Full support according with experiments 2, 3 and 4 from Section III.B (RBAC). As described before RBAC offers administrative benefits in assigning/revoking subjects and capabilities, i. e., with few steps. Partial support according with experiments 2, 3 and 4 from Section III.A (DAC) because is not possible to assign privileges to subject group. Notice, MS SQL Server has the concept of schema to minimize the number of steps.
How many steps are required for assigning subject groups and group relations or assigning object groups and group relations?	Full support according with experiments 2, 3 and 4 from Section III:B (RBAC). As described before RBAC offers administrative benefits in assigning/revoking subjects and capabilities, i. e., with few steps. No supported according with experiments 2, 3 and 4 from Section III.A (DAC) because is not possible to assign privileges to subject group. Notice, MS SQL Server has the concept of schema to minimize the number of steps.
How many steps are required for assigning privilege inheritance?	Full support according with experiments 2, 3 and 4 from Section III.B (RBAC). Notice, in RBAC case a subject X is assigned to database roleA, which has access privileges from database roleB, and then X automatically inherits all the privileges of B. No support according with experiments 2, 3 and 4 from Section III.A (DAC).

According to the analysis previously conducted, easy of privilege assignments function is fully supported by MS SQL Server 2012, when RBAC access control is used. When DAC access control is used, MS SQL Server partially supports easy of privilege assignments function.

For the delegation of administrative capabilities function from administration category, the item is analyzed in Table IV.

TABLE IV
DELEGATION OF ADMINISTRATIVE CAPABILITIES ITEMS

Metric Items to Evaluate	Analysis
Does the AC system allow policy administration delegation?	Full support according with experiment 6 from section III.A and III.B.

According to the analysis previously achieved, delegation of administrative capabilities function is fully supported by MS SQL Server 2012. For the least privilege principle support function from enforcement category, the items are analyzed in Table V.

TABLE V
LEAST PRIVILEGE PRINCIPLE SUPPORT ITEMS

Metric Items to Evaluate	Analysis
Is the AC system capable of enforcing the least privilege principle?	Full support according with experiment 1 from section III.A and III.B.
Does the AC system allow specifying least privilege via constraints?	No support.
Does the AC system allow specifying least privilege via other specifications?	No support.

According to the analysis previously achieved, least privilege principle function is partial supported by MS SQL Server 2012.

For the Separation of Duty (SoD) function from enforcement category, the items are analyzed in Table VI.

TABLE VI
SEPARATION OF DUTY (SOD) ITEMS

Metric Items to Evaluate	Analysis
Is the AC system capable of specifying Static SoD rules?	Full support according to role definition in MS SQL Server, i.e., when use RBAC access control.
Is the AC system capable of specifying Dynamic SoD rules?	Not supported since MS SQL Server does not support Chinese Wall policy.
Is the AC system capable of specifying Historical SoD rules?	Not supported since MS SQL Server does not support Clark-Wilson policy.

According to the analysis previously achieved, SoD function is partially supported by MS SQL Server 2012. Notice, MS SQL Server supports SoD when RBAC access control is used.

For the safety (confinements and constraints) function from enforcement category, the items are analyzed in Table VII.

TABLE VII
SAFETY (CONFINEMENTS AND CONSTRAINTS)

Metric Items to Evaluate	Analysis
Does the AC system provide safety check capabilities to prevent leaking of permissions?	Partial support according with experiments 3, 4 and 8 from Sections III.A and III.B. WITH GRANT clause and execution of stored procedures can leak permissions.

According to the analysis previously achieved, safety function is partially supported by MS SQL Server 2012.

For the conflict resolution or prevention function from enforcement category, the items are analyzed in Table VIII.

TABLE VIII
CONFLICT RESOLUTION OR PREVENTION ITEMS

Metric Items to Evaluate	Analysis
Is the AC system capable of preventing policy rule conflicts?	Full support according with experiments 5 and 6 from section III.A and III.B. DENY permission has precedence over GRANT permission.
Is the AC system capable of resolving conflict policy rules?	Full support according with experiments 5 and 6 from section III.A and III.B. DENY permission has precedence over GRANT permission.

According to the analysis previously achieved, conflict resolution function is full supported by MS SQL Server 2012. Finally, Table IX shows a comparative analysis of DAC and RBAC access control supported by MS SQL Server 2012 based on NIST functions described above.

TABLE IX
COMPARATIVE ANALYSIS OF DAC AND RBAC

NIST functions	DAC	RBAC
Auditing	Full support	Full support
Privileges / capabilities discovery	Partial support	Partial support
Ease of privileges assignments	Partial support	Full support
Delegation of administrative capabilities	Full support	Full support
Least privilege principle support	Partial support	Partial support
Separation of Duty	No support	Partial support
Safety (confinements and constraints)	Partial support	Partial support
Conflict resolution or prevention	Full support	Full support

According to the analysis previously achieved, the RBAC model has a slight advantage on the DAC model in the MS SQL Server 2012 context, considering ease of privileges assignments and separation of duty metrics.

V. RELATED WORK

The work in [3] analyzes and discuss RBAC in the context of DBMS products (Informix, Sybase and Oracle) based on i) role creation; ii) user role assignments and role propagation; iii) role activation; iv) creation of role hierarchies and constraints; and v) assignable privileges. In this paper some of the above criteria i), ii), v) are used to analyze and discuss access control models (DAC, MAC and RBAC) of a well-known DBMS: MS SQL Server 2012. The analysis and discussion have been improved using NISTIR 7874 criterion.

In [7] existing cloud based access control system were analyzed and evaluated using NIST access control systems evaluation criteria. Our work differs from this because we use a commercial database management system and the concept of discretionary and non-discretionary access control models.

The paper [1] presented the basic concepts of access control and investigated different issues to address the needs of an access control system. Also, the work in [9] discusses basic concepts about access control, showing the main characteristics of the discretionary, mandatory, and role-based access control policies along with their advantages and disadvantages. These two works were used to inspire the assessment proposed here; in particular the concepts about access control were the basis of our work.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, detailed analysis is performed for DAC and RBAC access control mechanisms in MS SQL Server 2012, a commercial DBMS. Our analysis shows that the MS SQL Server access control mechanisms target the main aspects of DAC and RBAC models. Also, some aspects were identified that can hamper the access control policy, such as procedures.

Based on the guidelines for access control system evaluation metrics (NIST 7874), RBAC model has a slight advantage on the DAC model in the MS SQL Server 2012 context. This smallest advantage is mainly due to privileges assignment issues, specially administrative. During the application of the evaluation metrics proposed by NIST, some difficulties have arisen. For example, it was not easy to apply the scale to the item related to the number of steps of ease of privileges assignments metric. Also, there is no available documentation with examples of practical application of NIST metrics.

For future work we will apply performance and support metrics to MS SQL Server; test another commercial DBMS and compare the results with MS SQL Server; use a real-world access control policy.

REFERENCES

[1] S. Vimercati, S. Foresti, P. Samarati, “Recent Advances in Access Control”, in Handbook of Database Security: Applications and Trends, (ed.) M., Gertz, S. Jajodia, Springer, 2008, pp. 1-22.

[2] V. C. Hu, D. F. Ferraiolo, D.R. Kuhn, “Assessment of Access Control Systems”, National Institute of Standards and Technology -NIST. Report NISTIR 7316, 2006.

[3] D. F. Ferraiolo, D. R. Kuhn, R. Chandramouli, “Role-Based Access Control”, Second Edition, Information Security and Privacy Series, (ed.) Rolf Oppliger, ArTech House, 2007.

[4] D. Ferraiolo, D. R. Kuhn, "Role-Based Access Control”, in Proceedings of the NIST-NSA National (USA) Computer Security Conference, 1992, pp. 554-563.

[5] K. Simmons, S. Carstarphen, “Pro SQL Server 2012 Administration”, Second Edition, Apress, 2012.

[6] R. Sandhu, E. Coyne, H. Feinstein, C. Youman, “Role-Based Access Control Models”. Computer Vol. 29, 2, 1996, pp. 38-47.

[7] Um-e-Ghazia, R. Masood, M.A. Shibli, “Comparative Analysis of Access Control Systems on Cloud”, in Proceedings of the 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012, pp.41-46.

[8] V. C. Hu, K. Scarfone, “Guidelines for Access Control System Evaluation Metrics”, National Institute of Standards and Technology - NIST. Report NISTIR 7874, 2012.

[9] E. Bertino, , R. Sandhu, “Database security - concepts, approaches, and challenges”, IEEE Transactions on Dependable and Secure Computing, vol.2, no.1, 2005, pp.2- 19.

[10] D. F. Ferraiolo, R. Sandhu, , S. Gavrila, D. R. Kuhn, “Proposed NIST Standard for Role-Based Access Control”, ACM Transactions on Information and Systems Security, Vol. 4, No. 3, 2001.